# Contents

TIFFDIFF Sorting Program	
TIFFDIFF Usage	4
TIFFDIFF Usage	5
TIFFDIFF Usage	6
TIFFDIFF Usage	7
TIFFDIFF Usage	8
Magix Movie Edit	9
Importing Images	
Adjusting Image Length	
Adjusting Image Length	
Frame Shuffling	13
Frame Shuffling	14
Checking Images for Transitions	15
Export	16
Appendix 2: Tiffdiff Source	
Appendix 2: Tiffdiff Source	19
Appendix 2: Batch File	20
Appendix 2: Parser	21
Appendix 2: Read tif File	22
Appendix 2: Image Raster	23
Appendix 2: Pixel Comp	24
Appendix 2: Pixel Comp	25
Output Batch	26
Appendix 2: Frame Insert	27
Appendix 1: Frame queue	

# **TIFFDIFF Sorting Program**

The video images now have duplicates and transitions in them. The images have to be copied over to a new directory without duplicates and transitions.

This can be done manually but it is pretty tedious and time consuming.

That is why I wrote a program that automates the process. The program makes mistakes here and there if the video is dark but generally works quite well.

The program is called TIFFDIFF. It takes the original images one by one and looks at the difference, pixel by pixel. If there is a sufficient difference, it assumes a transition. This way the program can learn where the transitions are in a sequence of images. It then marks the images in the center between transitions to be copied over to the destination directory.

The tiffdiff usage is pretty simple. Copy the following two programs to the directory where the video pictures reside:

1. tiffdiff.exe

2. run\_diff\_filter.bat You can obtain the program at: http://dl.dropbox.com/u/5667638/tiffdiff.zip

<b>W</b> UltraEdit	-32 - D:\stan\flicker_filter_diff\tiffdiff.c				- <u> </u>
Eile Edit Sea	arch Project View Format, Column Macro Advanced Window He	db			
D 🚅 🖬 🖬	🖨 🖪 🖁 Aα   ₩₂   H   📰   X 🗈 🛍   E ≡ ≡ 🔳   M M 🕯		] 🔁   💈   🤶	<b>k</b> ?	
Filter:	D:\stan\flicker_filter_diff\tiffdiff.c				
Open Files 💌	1 #include "tiffio.h"				
	2 #include <stdio.h></stdio.h>				
en Files	3 #include <string.h></string.h>				
D:\stan\flicker_	4				
	5 #define EXT "tif"				
	6 #define LINE_SIZE 80				
	s char in_file_name[] ="listing.txt";				
	10 char filter out file[] = "filter out bat"				
	li char nrevious nic name1[80]:				
	12 char previous pic name2[80];				
	13 char previous pic name3[80];				
	14 char previous pic_name4[80];				
	<pre>15 char previous_pic_name5[80];</pre>				
	<pre>16 char previous_pic_name6[80];</pre>				
	17 char previous_pic_name7[80];				
	18 char previous pic_name8[80];				
	20 cher previous pic names[00];				
	21 char previous pic name10[00];				
	22 char previous pic name12[80];				
	23 char previous pic name13[80];				
	24 char previous pic name14[80];				
	<pre>25 char previous_pic_name15[80];</pre>				
	<pre>26 char name_size_char[3];</pre>				
	•				
• •					
For Help, press l	=1	Ln 1, Col. 1, CW	UNIX	Mod: 11/26/2010 10:47:2 File Size: 9219	INS //

Open up the dos window: Start -> All Programs -> Accessories -> Command Prompt In the command window type the drive letter where your pictures reside followed by a colon and hit enter. Do a directory change by typing: cd dir\_path\_where\_pictures\_reside See example shown in the picture above. Type: run\_diff\_filter to invoke the batch program. The batch program creates destination directory (filter\_out\_dir). After that, the batch program runs the directory listing and stores it into a file (listing.txt) and then runs the tiffdiff program. The first thing that tiffdiff does is the directory listing cleanup. The directory listing has tif files in it but also

has some additional info such as time and date fields, the parent directory reference etc. Tiffdiff strips all of that extraneous information and keeps the tif file names only. A cleaned up listing is stored in listing\_out.txt file.

🗠 Command Prompt	×
C:>>d:	
D:\>cd \stan\flicker_filter_diff\pics	
D:\stan\flicker_filter_diff\pics>run_diff_filter	
D:\stan\flicker_filter_diff\pics}md filter_out_dir	
D:\stan\flicker_filter_diff\pics>dir 1>listing.txt	
D:\stan\flicker_filter_diff\pics>tiffdiff 30000 4 thres = 30000 frame_length = 4 picture name=0 picture name=0.0 picture name=0.03 picture name=0.033 picture name=0.03333 picture name=0.03333. picture name=0.03333. picture name=0.03333.ti picture name=0.03333.ti picture name=0.03333.tif picture name=0.03333.tif picture name=0.03333.tif picture name=0.03333.tif picture name=0.06633.tif picture name=0.06663.tif picture name=0.066666.tif picture name=0.066666.tif	
picture name=0.099996.tif picture name=0.099999.tif picture name=0.099999.tif	•

Tiffdiff then follows the index in listing\_out.txt and reads the tif pictures sequentially in the order that they are listed in the file. A pixel by pixel comparison is done between consecutive pictures as shown in the picture above. The difference is printed in the command window.

The difference is not run across all pixels, but instead three strips are used, one on the top, one in the middle and one on the bottom of the picture. The three difference values are printed as diff1, diff2, and diff3. If any of the diff values exceeds a threshold it is assumed that the picture has a transition.

As the program reads the consecutive pictures, the transitions are observed. The pictures in the middle between two transitions are considered as good candidates to be copied to the destination directory.

Command	Prompt			_	
diff1=80570	diff2=-61233	diff3=329491			
diff1=2120	diff2=370	diff3=-12019			
diff1=-876	diff2=16015	diff3=-8258			
diff1=-7639	diff2=30540	diff3=-3708	save file=	0.066666.tif	
diff1=6667	diff2=1057	diff3=-212			
diff1=9705	diff2=-679287	diff3=-3066	save file=	0.133332.tif	
diff1=-12749	diff2=1339009	diff3=-16108			
diff1=-8096	diff2=-2925	diff3=8451			
diff1=311	diff2=23018	diff3=-7886			
diff1=556	diff2=-11457	diff3=2095			
diff1=7257	diff2=-29936	diff3=-14435			
diff1=-13715	diff2=-1176799	diff3=11305			
	insert image=0.26	6664.tif			
save file=	0.333330.tif				
diff1=15809	diff2=1196321	diff3=6526			
diff1=-3871	diff2=-4910	diff3=-7605			
diff1=-2084	diff2=19109	diff3=-2559			
diff1=4689	diff2=-11368	diff3=516			
diff1=-6202	diff2=-13355	diff3=-4781			
diff1=7214	diff2=-1061949	diff3=30747			
20128-0000-007 - 0097675	insert image=0.46	6662.tif			
save file=	0.533328.tif				
diff1=-1351	diff2=1021099	diff3=-27063			
diff1=-4289	diff2=-24482	diff3=11659			
diff1=1060	diff2=24421	diff3=2706			
diff1=1264	diff2=-35870	diff3=-2864	save file=	0.666660.tif	
diff1=-1217	diff2=-10894	diff3=-2834			
diff1=-1013	diff2=530773	diff3=93394	save file=	0.733326.tif	
diff1=1661	diff2=-523130	diff3=-93680			
diff1=771	diff2=-12232	diff3=6109	1025-0127	1993 CONTRACTOR (1993)	
diff1=9061	diff2=34997	diff3=18496	save file=	0.833325.tif	
diff1=-4133	diff2=-71170	diff3=-3871			
diff1=5722	diff2=18287	diff3=2195		0.000000	
diff1=-3894	diff2=231638	diff3=225747	save file=	0.933324.tif	
diff1=10036	diff2=-297120	diff3=-211295			
diff1=-5201	diff2=-359	diff3=-89			
diff1=2443	diff2=11688	diff3=-5231			
diff1=7419	diff2=-14843	diff3=14092			
diff1=-4278	diff2=-8081	diff3=-2961			
diff1=-6327	diff2=89837	diff3=304579			
	insert image=1.06	6656.t1f			
save file=	1.133322.tlf	1.000 00000			
$a_{1ff1}=3267$	aiff2=-47983	alff3=-308297			
$a_{1ff1} = -496$	aitt2=-8719	alff3=-2497			
$a_{1ff1} = -2375$	aiff2=29282	aiff3=-18959			
$a_{1ff1} = -2282$	aitt2=-27559	aiff3=20338			
aiff1=13654	aitt2=4743	aiff3=6719			8

It is to be noted that tiffdiff does not actually copy the pictures. It enters the suitable picture names into the copy batch file (filter\_out.bat). Once the tiffdiff program is finished, the filter\_out.bat bach file is run automatically and does the actual copying. All that is left to be done is to go to the filter\_out\_dir and convert the pictures there into an avi format.

But before we do that let's have a look at two important parameters in the run\_diff\_filter batch file. Open up run\_diff\_filter.bat with a text editor. The file has the following content:

mkdir filter\_out\_dir dir /OD > listing.txt tiffdiff.exe 30000 4 filter\_out.bat

🗠 Command Prompt	_ 🗆 🗙
D:\stan\flicker_filter_diff\pics>copy 1.333320.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 1.466652.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 1.533318.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 1.699983.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 1.833315.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 1.966647.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.099899.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.199898.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.299897.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.433229.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.499895.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.633227.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.699893.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.833225.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics>copy 10.899891.tif filter_out_dir\*.* 1 file(s) copied.	
D:\stan\flicker_filter_diff\pics}	-

Look at the tiffdiff line. Tiffdiff is invoked with two parameters. The first one has a value of 30000 and the second one a value of 4. The first parameter is the difference threshold that we already discussed. The threshold is used to determine if the difference is large enough to classify the picture as a transition. You can change this value and run test videos with it. If the value is set too low, the program becomes more sensitive to changes and could detect differences caused by a camera vibration. Although the system is mechanically stable and solid, the projector motor, sounds etc will cause some differences in pictures. Another source of differences could be the camera auto exposure system. In any case do not set this value too low. If the threshold is set too high the program may skip some transitions. Some dark videos have very faint transitions and the program can miss them.

In that case you will end up with missed frames and jumpy video.

The second parameter is the step size measured by the number of pictures between the transitions.

Command Prompt	_ 🗆 🗙
C:\>d:	
D:\/cd \stan\flicker_filter_diff\pics	
D:\stan\flicker_filter_diff\pics>run_diff_filter	
D:\stan\flicker_filter_diff\pics>md filter_out_dir	
D:\stan\flicker_filter_diff\pics>dir 1>listing.txt	
D:\stan\flicker_filter_diff\pics>tiffdiff 30000 4 thres = 30000 frame_length = 4 picture name=0 picture name=0.03 picture name=0.033 picture name=0.03333 picture name=0.033333. picture name=0.033333.tif picture name=0.033333.tif picture name=0.033333.tif picture name=0.033333.tif picture name=0.063333.tif picture name=0.066633.tif picture name=0.066663.tif picture name=0.066666.tif picture name=0.096666.tif picture name=0.096666.tif picture name=0.099906.tif picture name=0.099990.tif picture name=0.099990.tif	

The program uses this parameter for cases where the transitions are not pronounced and are not detected by the program. The program keeps a picture count after the last transition. If the count exceeds the value of the second parameter, the program will force a transition. This way the number of skipped frames is reduced when video is dark and the transitions are faint. The count can be easily estimated just by looking at the pictures in thumbnail mode and counting the number between transitions. If the count is set wrong, the program will force transitions in wrong places and you will end up with some transitions in the video.

These transitions in the video can be manually deleted.



Transition Frame is smudged

### **Magix Movie Edit**

Once the tiffdiff program is finished with processing of the pictures, go to into the filter\_out\_dir destination folder.

This is where the tiffdiff program stored all processed pictures.

Run the Magix Movie Edit program.

In the filter\_out\_dir folder select all images and drag them into the video track area of the editor.



# **Importing Images**

s M G s M G s M G

The images will now appear on the edit track.

Select one of the images and right click the mouse.



# **Adjusting Image Length**

With one of the images selected and highlighted in orange, right click the mouse.

A menu pops up. Click on Change photo length.





# **Adjusting Image Length**

Change Image length to 00:00:02 for 8mm and super8 movies. Other types of movies may need a different number and it may need experimenting until the video playback is just right.

Click on Apply to all button. The edit track will show new length and the individual images will no longer be visible.

Click on the + button in the lower right corner. Keep clicking until individual images can be distinguished.

🕈 Movie Ed	dit Pro 15	Download	version - 2	010-10-0	2.MVP*											
MAGIX	66	📲 🗞 📢	File Edit	Effects	Window	Online Ta	asks He	ip 🕅		-	💽 Rec	ord 🛞	Edit		Burn	
00:00:25:24	2010	-10-02 *	00:00:25:2	4			_		_					_		×
				Impo	rt Fad	les Tit	le	Effects							COCC	NOH
	-AL			- <	> <b>€</b> 1≣	Option	15 👻	F:\mu	isic_transfer\i	marino_8r	nm\wedding	g\avi			• =	
	10	101			📲 🧰 mar	ino_8mm		▲	Name		Туре				Modified	A
- 0		50				anada_1983_1 parina	1987	툪ᄜ	a_marina1		.AVI	85,	443,072 9	/19/2010 12:	58:07 PM	
						arino makar	eka (	뮲	a_marina10		.AVI	81,	290,240 9	/24/2010 4:5	5:35 PM	
	1.000		122			witzerland	SKU	몸	a_marina11		.AVI	18,	543,872 9	/26/2010 2:5	6:31 PM	
		Species C.			• 🖬 t	est		■ ₩ ₩	a_marina12a		.AVI	/6,	413,952 9	0/20/2010 3:5	9:47 PM	
					🖌 🧰 v	redding			a marina12b		AVI	12, 66	094,200 1 450 944 1	0/2/2010 12:	22:02 PW	
2,24	add all	and all a	de al est		(	avi		₩ ma	a marina13		AVI	90	950.144 9	/26/2010 10:	13:12 PM	
00					۱.	pics		🗒 mal	a_marina14		.AVI	46.	803,968 9	/26/2010 10:	32:18 PM	
TH					🕨 💼 mid	i.		🔻 🛱 mal	a marina15		.AVI	55.	141.888 9	/28/2010 8:2	6:10 PM	
						)	•									▶
		⊛					5	C ×	T 🕨	Ð	90 8	- Th + 10	* - {=+}	- F	486	×
	3	1		00:25:24			}									
		00:00:00	00:05:00	00:10:00	00:15:00	00;20:00 1 .0	9:25:00	00;39:00	.00:35:00	0;49:00	00:45:00	00:59:000	0x55:00	01:00:00	01:05:00	
SMÉ	6															1
Tr	rack:															
S M C	ñ															
5 IVI 6	<u> </u>															
11																-
SMÉ	8															
SM	<del>û</del>															
The second s																
SMO	<b>n</b>	-														-1
2 IVI 6	- 5															
SM	6															
i ii																_
SMÉ	8 7															
1																
SM																
- Tr	nini -															+
		< 00												D Zoom 🔻	<b>4</b> - <b>+</b>	+
CPU:																

Display duration for the current object	×
×	→ 00:00:02 sec
Apply to selected range	
OK Apply to all	Cancel

# **Frame Shuffling**

Click on the first frame and observe the monitor window. Click on the second frame next and check if the scenery in the monitor changes a lot. If it does then the last frame has been inserted in the first spot. This program does that a lot and you have to keep an eye on it.

If the last frame is at the beginning you will have to move it to the end.

🕈 Mov	ie Edit I	Pro 15	Download	version -	2010-1	0-02.M	VP*															×
MAG	IX 🛛		🖷 🗞 🔖							Help (	P				•	Record	8	Edi	t (	0	Burn	
00:00:0	0:03	2010	10-02 *	00:00:25	× :24	_	-		with a					_	_		_		_		0100	× ०म
	100.00		And again			ipon 4	raues	Opt	ions	- Eneci	F:\mus	ic trans	fer\ma	rino 8r	nm\wee	ding\avi				_	==	
	A	K			F	40	marin	o_8mm				– Na	me		Т	/pe	Size		_	Mo	dified	
	10		1	前非			🕨 💼 can	ada_1983	_1987	2		_marina	1		.A	vi	85,	443,072	9/19/201	0 12:58	07 PM	
			a succession	×			<ul> <li>ma</li> <li>ma</li> </ul>	rina rino_mal	arska	1000	긒 mala 그 mala	_marina _marina	10 11		.A. .A	vi vi	81, 38,	290,240 1 543,872 1	9/24/201 9/26/201	0 4:55: 0 2:56:	5 PM 1 PM	
		6	ALC: N	4			Swi Swi Swi Swi Swi Swi Swi Swi Swi Swi	tzerland			nala	_marina	12		.A	vi	76,	413,952	9/26/201	0 3:59:4	7 PM	
							week	dding			nala mala	_marina _marina	12a 12b		A. A.	vi vi	12, 66,	894,208 450,944	10/2/2010 10/2/2010	0 12:22 0 12:26	02 PM 51 PM	
24	1.1.1.1	15 301		يا د باي				avi nics		Ş		_marina	13		.A	л	90,	950,144	9/26/201	0 10:13	12 PM	
	} {+			{ <b>&gt;</b> }		1	midi	, ici		•	mala	marina	14 15		.A.	vi VI	46, 55.	803,968 141.888	9/26/201	0 10:32 0 8:26:1	0 PM	
	88 📰		æ.,			-	-	-	ן ר	<u>}</u>	×	т		5	9 C	2 =	<i>11</i>	• ▼ {=±}	Ţ	M	581.	×
		3									00:2	25:24				<u> </u>				-		
	M 0		00:00:00	dante	rtru	The los	0101:00.	Hark	n de	111	00:02:0	Î I I I			hud	00:03:00	<u>eler</u> i	here	<u>ri'ri</u>	. 00:0	1:00.1.1	
51	VI izi Track:	1	<b>Parts</b>			3335	222	1155	553		- 1 )	111	333	311	111			111		111	111	Π
S I	мы																					
S I	NB																					
SI	MА																					
in the second	Tende																					
SI	VI 🔐																					
5 1	ИВ																					
S I	N B																					
S I	м																					
	Tetele																		D Zo	om v ৰ		•
CPL	J:							_														
				_								in the			100000	<u> </u>					_	×
	88 🖮			15/2/1		_	_	_	2	C	×	T	-	Э	90	Ċ	<b>∖</b> -∥	• - {=*)	*	×	111	
		30		25:00.1	r 	i:@Q		27:99	1	28:00		.00:22:00	2		0:00	R	0:31:00		00:82:90.			
5 1	VI 🔓 Track:	1																				20
S N	M 🔒																					
5 1	N B																					
S N	N A																					
5.0	inada A A																					
	Track																					
SI	и 🔒																					
S N	a N																					
S N	N 8																					101
			< 600																> Zo	om v 🔹	-	+

# **Frame Shuffling**

Click and drag it to the end behind the last frame.

Now go back to the beginning and hit the shift key on the keyboard and click on the first frame. This will cause all of the frames to be highlighted (selected). Let go of the shift key and drag the frames to the left until the frames line up with the start of the track.

	8			⊛ -		り	C	×	T 🕨	C 1	90	8	<b>∖</b> -∥	<b>▼</b> {=*} <b>▼</b>	I	481	×□
			6		0.00.00	0.02.00	a	00:25:2	24	00-05-00		00.05.00		00-07-00		0.02.00	
5	м	G Track:	1														
S	м	6															
5	м	<b>G</b>															
	м	6															
5	м	6															
S	м	6															
5	М																
s	м	6															
5			1	< m	1					<u>.</u>					Zoom v		+

		& -	っ	C	×	Т		Э	90	ð	$\neg_{t^{\star}}$	1-1	<b>-</b> *}▼	M	488	×
	6				00:2	5:24				00.05			7.00		2.00	
S M 🔒 Track:	1						1									12
S M 🔒																
S M 🔒																
S M 🔒																
SM 🔒																
S M 🔒																
S M 🔒																_
SMG																- +
		< m												Zoom 👻 🗟		

# **Checking Images for Transitions**

Press the right arrow on the keyboard and observe the monitor window.

The program will start scanning through the images and will appear as a video running at a slow rate.

Observe for transitions. They will appear as a flicker in the video.

Once a transition is detected, stop the scan and use the left and right arrow keys until the transition appears in the monitor.

The image will be highlighted in orange in the edit track.

Press the erase icon to erase the image.

A gap will appear in the track. Select all of the images to the right of the gap and drag them to the left until the gap is covered. Do not go too far to the left to avoid image overlapping.

Repeat the process until the end of the track.

You are now almost done.



#### **Export**

Now export the movie as AVI file. Obviously, other formats are possible depending on your preferences.

Magix Movie Edit can also burn the dvd. If that is your preferred format then just press the Burn button on the top menu and follow the prompts.

And that concludes this chapter. The videos produced using this method will have very little flicker but the resolution will still be low and the colors will look washed out.

Further improvements can be obtained by grabbing the image directly from the projector without any screens in between. More on that and other methods will be covered in the following chapters.

MACIX		File File Filects Western	Online Task	s Hela D	• • • • • • •	4 (2) 100							
0:00:00:03	2010-10-02 *	New project Load project Save project Save project as	Ctrl+N Ctrl+O Ctrl+S Shift+S	Effects Fi\music.transfer\marin	no 8mm\wedding\#	s Rmm\weddfina\avi							
		Manage movies		A Name	Туре	Size	Modified						
		Record audio/image/video Import audio CD track(s) Twain scanner	c c	mala_marina1 mala_marina10 mala_marina11	IVA. IVA. IVA.	85,443,072 9/15 81,290,240 9/24 38,543,872 9/26	9/2010 12:58:07 PM 9/2010 4:55:35 PM 9/2010 2:56:31 PM						
24 1. 1		Export movie Play video/audio Batch conversion Bum @/DVD Internet	H Alt+S	Video as AVL Video as DV-AVL Video as MPEC Video as MACIX video Video as QuickTime movie	Alt+A A Alt+8 A Alt+C A Alt+C A Alt+C A Alt+C A	76,413,952 9/26 12,894,208 10/2 66,450,944 10/2 90,950,144 9/26 46,803,968 9/26	6/2010 3:59:47 PM 2/2010 12:22:02 PM 2/2010 12:26:51 PM 6/2010 10:13:12 PM 6/2010 10:32:18 PM						
	θ+	Backup copy Load backup project Clean-up wizard	Alt+O Ctrl+Alt+G	Video as uncompressed video Video as Motion/PEC AVI Windows Media export Real/Media export	Alt+U Alt+F Alt+G	55.141.888 9/28	5/2010 8:26:10 PM						
-		MovieShow Maker Soundtrack Maker	Ctrl+Shift+M W	Video as MPEG-4	Alt+K								
-	Teores in	Settings		Audio as wave	Alt+H 2006.00	Control of	1.1. 10.000000						
S M G S M G	. 1	Exit	Alt+ <b>F</b> 4	Single frame as BMP Single frame as JPEG Animated GIP	Alt+N Alt+N								
S M 🔒													
S M 🔒													
S M 🔒													
SMB													
SMA													

AVI export	×
Presets	
User defined:	
Export settings	
Resolution:         1920x1080          Im           Frame rate:         15.00	terlaced mode: Progressive
Aspect ratio: 16:9	Video
File	\$5.10
F:\music_transfer\marino_8mm\wedding\avi\20	010-10-02.AVI
Overwrite file without a warning message	
Other	
<ul> <li>Only export selected range</li> <li>Use anti-flicker filter</li> <li>Shut down PC automatically after successful ex</li> </ul>	port
Play after export: Don't play out Target path:	
Reset Help	OK Cancel

I compiled a short program to do the frame difference analysis. I called the program tiffdiff.c.

The program uses the libtiff library to open up the tiff files and to do the pixel extraction.

Libtiff library can be obtained form : http://www.libtiff.org/

Download the latest version and extract it in a local folder of your computer. In order to compile the library you will need version 6 of Microsoft Visual Studio. The library can be compiled from a command line (dos window). Open up dos window and go to the library top directory. Run the following commands:

nmake /f makefile.vc clean

nmake /f makefile.vc

The library will compile and produce the output as shown in the picture.

🖾 Command Prompt
supplied tiffcrop.c(4640) : warning C4761: integral size mismatch in argument; conversion
supplied tiffcrop.c(4112) : warning C4761: integral size mismatch in argument; conversion supplied
tiffcrop.c(4114) : warning C4761: integral size mismatch in argument; conversion
tiffcrop.c(4115) : warning C4761: integral size mismatch in argument; conversion supplied
LINK : warning LNK4098: defaultlib "LIBC" conflicts with use of other libs; use NODEFAULTLIB:library
cl /nologo /OX /MD /EHsc /W3 /D_CRT_SECURE_NO_DEPRECATE -1libtiff -1. .port -DNEED_LIBPORT -DAUOID_WIN32_FILEIO -DCHECK_JPEG_YCBCR_SUBSAMPLING -DDEFA ULT_EXTRASAMPLE_AS_ALPHA -DSTRIPCHOP_DEFAULT=TIFF_STRIPCHOP -DSTRIP_SIZE_DEFAULT =8192 -DJPEG_SUPPORT -DOJPEG_SUPPORT -DLOGLUU_SUPPORT -DNEXT_SUPPORT -DTHUNDER_S UPPORT -DLZW_SUPPORT -DPACKBITS_SUPPORT -DCCITT_SUPPORT -DNEXT_SUPPORT -OTHUNDER_S UPPORT_LISSEMSB tiffdither.c d:/public/libjpeg/jpeg-8b//libjpeg.libNport \libport.lib\libtiff\libtiff.lib
LINK: warning LNK4098: defaultlib "LIBC" conflicts with use of other libs; use /NODEFAULTLIB:library
cl /nologo /Óx /MD /EHsc /W3 /D_CRT_SECURE_NO_DEPRECATE -I\libtiff -I. .\port -DNEED_LIBPORT -DAUOID_WIN32_FILEIO -DCHECK_JPEG_YCBCR_SUBSAMPLING -DDEFA ULT EXTRASAMPLE AS ALPHA -DSTRIPCHOP_DEFAULT=TIFF_STRIPCHOP -DSTRIP_SIZE_DEFAULT =8192 -DJPEG_SUPPORT -DOJPEG_SUPPORT -DLOGLUU_SUPPORT -DNEXT_SUPPORT -DTHUNDER_S UPPORT -DLZW_SUPPORT -DPACKBITS_SUPPORT -DCCITT_SUPPORT -DTIF_PLATFORM_CONSOLE - DFILLODER_LSB2MSB tiffmedian.c d:/public/libjpeg/jpeg-8b//libjpeg.lib\port \libport.lib\libtiff.lib
tiffmedian.c LINK : warning LNK4098: defaultlib "LIBC" conflicts with use of other libs; use
<pre>/NOBEFHULLIB:TIDPAPY cl/nologo /Ox/MD/EHsc/W3/D_CRT_SECURE_NO_DEPRECATE -1\libtiff -1. .\port -DNEED_LIBPORT -DAUOID_WIN32_FILEIO -DCHECK_JPEG_YCBCR_SUBSAMPLING -DDEFA ULT_EXTRASAMPLE_AS_ALPHA -DSTRIPCHOP_DEPAULT=TIFF_STRIPCHOP -DSTRIP_SIZE_DEFAULT =8192 -DJPEG_SUPPORT -DOJPEG_SUPPORT -DLOGLUU_SUPPORT -DNEXT_SUPPORT -DTHUNDER_S UPPORT -DLZW_SUPPORT -DPACKBITS_SUPPORT -DCCITT_SUPPORT -DTIF_PLATFORM_CONSOLE - DFILLODER_LSE2MSB tiffset.c d:/public/libjpeg/jpeg-8b//libjpeg.lib\port\li bport.lib\libtiff\libtiff.lib</pre>
LINK : warning LNK4098: defaultlib "LIBC" conflicts with use of other libs; use
cl/nologo/Ox/MD/EHsc/W3/D_CRT_SECURE_NO_DEPRECATE -I\libtiff -I. \port -DNEED_LIBPORT -DAUOID_WIN32_FILEIO -DCHECK_JPEG_YCBCR_SUBSAMPLING -DDEFA ULT_EXTRASAMPLE_AS_ALPHA -DSTRIPCHOP_DEFAULT=TIFF_STRIPCHOP -DSTRIP_SIZE_DEPAULT =8192 -DJPEG_SUPPORT -DOJPEG_SUPPORT -DLOCLUU_SUPPORT -DNEXT_SUPPORT -DTHUNDER_S UPPORT -DLZW_SUPPORT -DOJACKBITS_SUPPORT -DCCITT_SUPPORT -DTIF_PLATFORM_CONSOLE - DFILLODER_LSB2MSB_tiffsplit.c_d:/public/libjpeg/jpeg-8b//libjpeg.lib\port libport.lib\libtiff\libtiff.lib tiffsplit.c
LINK : warning LNK4098: defaultlib "LIBC" conflicts with use of other libs; use /NODEFAULILB:library cd
D:\stan\flicker_filter_diff>

# **Appendix 2: Tiffdiff Source**

The library object gets created in with the following path: libtiff\libtiff.lib The application then can be linked to the library by specifying this can during the application compilation process. For example, tiffdiff can be compiled with this library by running the following

command:

cl tiffdiff.c /MDd -Ilibtiff libtiff.lib tiffdiff test1.TIF

The /MDd option tells the compiler to link in libtiff library at location: libtiff\libtiff.lib

Once compiled this way the application can use all of the library calls.

D:\stan\flicker\_filter\_diff>cl tiffdiff.c /MDd -Ilibtiff libtiff\libtiff.lib Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8168 for 80x86 Copyright (C) Microsoft Corp 1984-1998. All rights reserved. tiffdiff.c tiffdiff.c(105) : warning C4552: '>=' : operator has no effect; expected operato r with side-effect Microsoft (R) Incremental Linker Version 6.00.8168 Copyright (C) Microsoft Corp 1992-1998. All rights reserved. /out:tiffdiff.exe tiffdiff.exe tiffdiff.obj libtiff\libtiff.lib LINK : warning LNK4098: defaultlib "MSUCRT" conflicts with use of other libs; use /NODEFAULTLIB:library LINK : warning LNK4098: defaultlib "LIBC" conflicts with use of other libs; use /NODEFAULTLIB:library D:\stan\flicker\_filter\_diff>\_

# **Appendix 2: Tiffdiff Source**

The tiffdiff source starts with

main(int argc, char\* argv[])

call with two arguments passed in from the command line. The first argument is the difference threshold and the second argument is frame length.

These two arguments were discussed in previous chapters already. The threshold sets the tool differentiation threshold and the frame length tells the tool how to insert the frames if the frames are missed.

The tool opens up three files:

in\_file\_name
out\_file\_name
filter\_out\_file

#### The files have the following declarations:

char in\_file\_name[] ="listing.txt"; char out\_file\_name[] ="listing\_out.txt"; char filter\_out\_file[] = "filter\_out.bat";

```
(≝) tiffdiff.c - WordPad
<u>File Edit View Insert Format H</u>elp
 D 😅 🖬 🎒 🔖 👫 🐴 👗 🖻 🛍 🗠 🦉
  int offset_counter;
 main(int argc, char* argv[])
 TIFF* tif:
  FILE *fp,*fp1,*fp2;
  char input_line[LINE_SIZE];
  char tiff_file_name[LINE_SIZE];
  char picture name[] = "
                                              ";
  int count;
  char ext text[]="nul";
  sscanf(argv[1], "%d", &thres);
  printf("thres = %d\n", thres);
  sscanf(argv[2], "%d", &frame length);
  printf("frame length = %d\n", frame length);
  frame length and half = frame_length_and_half_lookup[frame_length];
        if ((fp = fopen(in_file_name,"r")) == NULL)
        {
              printf("Can't open %s\n", in_file_name);
        }
        if ((fp1 = fopen(out file name, "w")) == NULL)
        {
              printf("Can't open %s\n", out_file_name);
        3
        if ((fp2 = fopen(filter_out_file,"w")) == NULL)
        {
              printf("Can't open %s\n", out_file_name);
        }
For Help, press F1
```

# **Appendix 2: Batch File**

The listing.txt file gets generated by "run\_diff\_filter.bat" batch file. The batch file is pretty simple and here is what it looks like:

```
mkdir filter_out_dir
dir /OD > listing.txt
tiffdiff.exe 30000 4
filter_out.bat
```

The second line of the batch file creates the listing.txt file. It essentially runs the "dir" command and reroutes the output listing of the directory into the listing.txt file. The listing\_out.txt file is a temporary storage where tiffdiff will send its output. Tiffdiff has to preprocess the listings.txt file because the directory listing is cluttered with extra stuff that the program does not need as shown in the picture.

Command Prompt		
Volume Serial Number	is A4C1-DD9E	
Divectory of 7:		
Directory of 2.		
Z • N Nd in		
Volume in drive Z is I	Main_archiv	
Volume Serial Number :	is A4C1-DD9E	
Directory of Z:\		
	To be a second	
12/26/2010 10:59 HM 12/26/2010 10:59 AM	<dir></dir>	
10/01/2010 10:41 PM	2,581,466 153.598464.tif	
10/01/2010 10:41 PM	2,513,096 153.631797.tif	
10/01/2010 10:41 PM	2,581,664 153.698463.tif	
10/01/2010 10:41 PM	2,582,670 153.731796.tif	
10/01/2010 10:41 PM 10/01/2010 10:41 PM	2,592,178 153.765129.t1f 2 605 632 153 798462 fif	
10/01/2010 10:41 PM	2,620,698 153.831795.tif	
10/01/2010 10:41 PM	2,492,530 153.865128.tif	
10/01/2010 10:41 PM	2,574,078 153.931794.tif	
10/01/2010 10:41 PM	2,551,002 153.965127.tif	
10/01/2010 10:41 PM 10/01/2010 10:41 PM	2,381,550 153.998460.tlf 2.251.346 154.031793.tif	
10/01/2010 10:41 PM	2,276,472 154.065126.tif	
10/01/2010 10:41 PM	2,263,026 154.098459.tif	
10/01/2010 10:41 PM	2,238,308 154.165125.tif	
10/01/2010 10:41 PM	2,217,998 154.198458.tif	
10/01/2010 10:41 PM 10/01/2010 10:41 PM	2,225,558 154.231791.t1f 2.030.404 154.265124.tif	
10/01/2010 10:41 PM	2,200,636 154.298457.tif	
10/01/2010 10:41 PM	2,188,372 154.331790.tif 2 199 670 154 365123 tif	
10/01/2010 10:41 PM	2,163,794 154.398456.tif	
10/01/2010 10:41 PM	2,215,590 154.431789.tif	
10/01/2010 10:41 PM	2,231,182 154.465122.tlf 2.270.798 154.498455.tif	
10/01/2010 10:41 PM	2,271,950 154.531788.tif	
10/01/2010 10:41 PM 10/01/2010 10:41 PM	2,260,098 154.565121.t1f 2 263 142 154 598454 fif	
10/01/2010 10:41 PM	2,156,722 154.631787.tif	
10/01/2010 10:41 PM	2,306,826 154.665120.tif	
10/01/2010 10:41 PM	2,298,260 154.731786.tif	
10/01/2010 10:41 PM	2,085,936 154.765119.tif	
10/01/2010 10:41 PM	2,303,666 154.778452.tlf 2.330.622 154.831785.tlf	
10/01/2010 10:41 PM	2,299,638 154.865118.tif	
10/01/2010 10:41 PM 10/01/2010 10:41 PM	2,175,548 154.898451.t1f 2.376.332 154.931784.tif	
10/01/2010 10:41 PM	2,339,090 154.965117.tif	
10/01/2010 10:41 PM	2,343,534 154.998450.tif 2 202 712 155 031783 tif	
10/01/2010 10:41 PM	2,292,550 155.065116.tif	
10/01/2010 10:41 PM	2,290,144 155.098449.tif	
10/01/2010 10:41 PM	2.337.918 155.165115.tif	
10/01/2010 10:41 PM	2,377,228 155.198448.tif	
10/01/2010 10:41 PM	2,374,936 155.231781.tif 2 333 662 155 265114 fif	
10/01/2010 10:41 PM	2,317,452 155.298447.tif	
10/01/2010 10:41 PM	2,388,026 155.331780.tif	
10/01/2010 10:41 PM	2,230,396 155.398446.tif	
10/01/2010 10:41 PM	2,306,016 155.431779.tif	
10/01/2010 10:41 PM	2,325,356 155.465112.tlf	

### **Appendix 2: Parser**

The parser searches the input file for the .tif extension.

Once it fonds the .tiff extension it flags that as the end of the name. After that the program backs off the character pointer until it finds a space 0x20. The space represents the beginning of the name string. So now we know where the beginning and the end of the name within the input line. With that the program reads the name string and stores it into the listing\_out.txt file. After that the program reads the next line and store the next name and so on until it reaches the end of the input file. This completes the parse section of the program.

```
20
       // skip header
81
      for (1=0; 1<5; 1++)
82 E
02
      fgets(input_line, LINE_SIZE, fp);
24
35
-
      while(fgets(input_line, sizeof(input_line), fp))
19
      // Check position of ".tiff" picture extention
90
91
        for (i=0; i<(sizeof(input line)); i++)</pre>
92 E
52
          for(j=0; j<3; j++)
94日
95
          ext_text[j] = input_line[i+j];
56
97
            printf("extention = %s\n",ext_text);
          if( stromp(ext_text, EXT) == 0)
98
99日
            printf("found extention = %s\n",ext text);
100
            printf("i= = %d\n",1):
          ext_pointer = i;
102
            // back off to find the begining of file name
103
104
            for (n=1-1;n--;n \ge 0)
101
LOEE
              if (input_line[n] == 0x20)
              // found the beginning
101
              begin_pointer = n;
              break:
              if (n == 0)
1148
              printf("problem with picture name - check the listings file\n");
116
              break;
11E
            1 = 0;
              for(k=begin_pointer+1: k < ext_pointer+3: k++)</pre>
121
              6
               printf("begin pointer =%d\n",begin_pointer);
128
                 printf("end pointer =%d\n",ext_pointer);
              picture_name[1] =input_line[k];
126
              printf("picture name=%s\n", picture_name);
127
              1++;
28
125
           picture name[1]=0;
138
            fputs(picture name, fpl);
            fputs("\n", fpi);
131
          2
132
        5
134
135
```

# **Appendix 2: Read tif File**

The next step is to use the listing\_out.txt file. The listing\_out.txt containing frame file names, one name per line. The program read the first name and then opens up the file with that name.

The program uses the libtiff TIFFOpen() call to open up the tiff file and create the file pointer tif.

The pointer is then used in subsequent calls to determine the size of the image and to get the raster.

The raster is a two dimensional array with the actual image pixel content.

```
149
       offset_counter = 0;
150
       previous_pic_ok = 0;
151
       while(fgets(input_line,sizeof(input_line), fp1))
152 -
       1
153
     // printf("input_line_size=%d\n",strlen(input_line));
 154
 155
       line size = name size char[0];
 156 // printf("line_size=%d\n",line_size);
157
       input_line[strlen(input_line) - 1] =0;
 158
159
160
161
       tif = TIFFOpen(input line, "r");
```

# **Appendix 2: Image Raster**

Each element of the raster array consists of three pixels: 1. red 2 green 3. yellow

The pixels can be obtained through TIFFGetR(), TIFFGetG(), and TIFFGetB() calls as shown in the picture.

Once read, the pixels get stored into 3 temp arrays:

r\_array[]

g\_array[]

b\_array[]

The temp arrays are used to compare the pixels between the current image and the previous image.

162		if (tif)
163	1	1
164		
165		uint32 w, h;
166		size_t npixels;
167		uint32* raster;
168		char R,G,B;
169	11	<pre>printf("I am here\n");</pre>
170		<pre>TIFFGetField(tif, TIFFTAG_IMAGEWIDTH, &amp;w);</pre>
171		<pre>TIFFGetField(tif, TIFFTAG_IMAGELENGTH, &amp;h);</pre>
172		npixels = w * h;
173		<pre>raster = (uint32*) _TIFFmalloc(npixels * sizeof (uint32));</pre>
174		if (raster != NULL)
175 -	1	4
176		<pre>if (TIFFReadRGBAImage(tif, w, h, raster, 0))</pre>
177 -	1	1
178	11	<pre>printf("pixel count=%d\n", npixels);</pre>
179		diff1 = 0;
180		diff2 = 0;
181		diff3 = 0;
182		<pre>for(i=0; i &lt; (w*5); i++)</pre>
183 🚍	1	
184		<pre>R=(char )TIFFGetR(raster[i]);</pre>
185		<pre>G=(char )TIFFGetG(raster[i]);</pre>
186		<pre>B=(char )TIFFGetB(raster[i]);</pre>

# **Appendix 2: Pixel Comp**

The current frame pixels R, G, and B are compared against the previous frame pixels stored in r\_array, g\_array, and b\_array.

The pixel difference is the sum of r, g, and b differences. The differences for all of the pixels contained within a 5 line strip are then added together.

The process is then repeated for two more 5 line strips.

The first strip is located at the bottom of the image, the second strip in the middle and the third strip at the top.

The reason for using three strips is to do with the nature of the transition. It is not very common that the transition is spread over the whole image. Very frequently it appears as a thin strip at the top or bottom of the picture. Doing the comparison over the whole picture in this case would not catch the transition.

11	STAN\music_t	transfer\8mm_video_transfer\flicker_filter_diff\tiffdiff.c	X
1	0	, 10,	
184		<pre>R=(char )TIFFGetR(raster[i]);</pre>	^
185		<pre>G=(char )TIFFGetG(raster[i]);</pre>	
186		<pre>B=(char )TIFFGetB(raster[i]);</pre>	
187		<pre>diff1_r= ((R) - r_array[i]);</pre>	
188		diff1_g= ((G) - g_array[i]);	
189		<pre>diff1_b= ((B) - b_array[i]);</pre>	
190	11	<pre>printf("R=%d\n",R);</pre>	
191	11	<pre>printf("i=%d\n",i);</pre>	
192		$r_array[i] = (R);$	
193		$g_{array}[i] = (G);$	
194		<pre>b_array[i] = (B);</pre>	
195		diff1 = diff1 + diff1_r + diff1_g + diff1_b;	
196		3	
197			
198		for $(i=(w^{h}/2); i < ((w^{h}/2)+w^{s}); i^{+})$	
199		£	
200		<pre>R=(char )TIFFGetR(raster[i]);</pre>	
201		<pre>G=(char )TIFFGetG(raster[i]);</pre>	E
202		<pre>B=(char )TIFFGetB(raster[i]);</pre>	
203		<pre>diff2_r= ((R) - r_array[i]);</pre>	
204		diff2_g= ((G) - g_array[i]);	
205		diff2 b= ((B) - b array[i]);	
206	11	<pre>printf("R=%d\n",R);</pre>	
207	11	<pre>printf("i=%d\n",i);</pre>	
208		$r_{array}[i] = (R);$	
209		$g_{array}[i] = (G);$	
210		<pre>b_array[i] = (B);</pre>	
211		diff2 = diff2 + diff2_r + diff2_g + diff2_b;	
212	- 11	3	
213			
214		for $(i=((w*h)-(w*5)); i < (w*h); i++)$	
215	5		
216		<pre>R=(char )TIFFGetR(raster[i]);</pre>	+
•		m	•
-			

# **Appendix 2: Pixel Comp**

The diffs are done for each strip and then compared against the threshold for each strip independently.

If all of the strips are below threshold the frame is considered to be good without transitions and previous\_pic\_ok flag is set.

If one or more strips are above the threshold, and previous\_pic\_ok is set, this indicates that the current frame has a transition.

If the frame has a transition in it and the previous frame was good then one of the previous good frames has to be stored into destination filter\_out\_dir folder.

The program keeps track of the frevious frames by using a queue. The actual frame that gets stored is not the previous frame but the one before that (previous\_pic\_name2). The previous frame could still have some effects of transition since it is so close to it.

🧕 \\ST/	V\music_transfer\8mm_video_transfer\flicker_filter_diff\tiffdiff.c
Q	
282	if( ((abs(diff1)) < thres) &&((abs(diff2)) < thres) &&((abs(diff3)) < thr
283 -	( )
284	<pre>previous_pic_ok = 1;</pre>
285	<pre>printf("\n");</pre>
286	
287 -	3
288	else
289	£
290	printf("good pic followed by bad pic\n");
291	<pre>printf("previous pic ok =%d\n", previous_pic_ok);</pre>
292	if (previous pic ok == 1)
293	ξ
294	<pre>printf("save file=\t");</pre>
295	<pre>printf("%s\n", previous_pic_name2);</pre>
296	<pre>fputs("copy ", fp2);</pre>
297	<pre>fputs(previous_pic_name2, fp2);</pre>
298	<pre>fputs(" filter_out_dir\\*.*\n",fp2);</pre>
299	<pre>printf("previous pic name=%s\n",previous_pic_name);</pre>
300	offset_counter = 0;
301 -	3
302	else
303 🚍	
304 )	offset_counter++;
305	<pre>printf("\n");</pre>
306 -	3
307	<pre>if(abs(diffl) &gt; thres)</pre>
308 /	<pre>printf("\n");</pre>
309	previous_pic_ok = 0;
310 -	) ····
311 -	<b>)</b>
312	_TIFFfree(raster);
313 -	

### **Output Batch**

So, it makes sense to back off another frame. It should be mentioned that the program does not do the actual copy of the frames to the destination folder, but instead it lists the frames to be copied in filter\_out.bat batch file. The batch file gets built on a fly as the program goes through all frames.

Once the tiffdiff program is finished, the main batch file runs the filter\_out batch. The filter\_out batch does the actual copy of all good frames into the destination dir. If the difference is larger than threshold and the previous frame was bad, the program assumes a transition spanning over multiple frames and essentially does not do anything, it just goes to the next frame.



# **Appendix 2: Frame Insert**

Depending on the threshold value and the frame content, the program can miss a frame transition from time to time. In order to prevent this from happening, a frame offset counter (offset\_counter) was implemented . The counter runs the frame count offset from the last transition. If the frame count exceeds the frame length by 50% a missed transition is assumed and the program forces a frame insert. The frame that gets inserted is picked from the frame queue so that it is half way between the last missed transition and the transition before that. This generally work pretty well and improves the program performance with dark scenes.

Market STAN/music_tr	ansfer\8mm_video_transfer\flicker_filter_diff\tiffdiff.c*
Q	1,0,
233	offset_counter++;
234	<pre>if (offset_counter == frame_length_and_half)</pre>
235 -	(
236	<pre>switch(frame_length)</pre>
237	1
238	case 3:
239	<pre>fputs("copy ", fp2);</pre>
240	fputs(previous pic name3, fp2);
241	<pre>fputs(" filter out dir\\*.*\n",fp2);</pre>
242	break;
243	case 4:
244	<pre>fputs("copy ", fp2);</pre>
245	<pre>fputs(previous_pic_name4, fp2);</pre>
246	<pre>fputs(" filter_out_dir\\*.*\n",fp2);</pre>
247	break;
248	case 5:
249	<pre>fputs("copy ", fp2);</pre>
250	<pre>fputs(previous_pic_name5, fp2);</pre>
251	<pre>fputs(" filter_out_dir\\*.*\n",fp2);</pre>
252	break;
253	case 6:
254	<pre>fputs("copy ", fp2);</pre>
255	<pre>fputs(previous_pic_name6, fp2);</pre>
256	<pre>fputs(" filter_out_dir\\*.*\n",fp2);</pre>
257	break;
258	case 7:
259	<pre>fputs("copy ", fp2);</pre>
260	<pre>fputs(previous_pic_name7, fp2);</pre>
261	<pre>fputs(" filter_out_dir\\*.*\n",fp2);</pre>
2.62	break;
263	case 8:
264	<pre>fputs("copy ", fp2);</pre>
265	<pre>fputs(previous_pic_name8, fp2);</pre>
•	
and the second s	

# **Appendix 1: Frame queue**

The program maintains a queue of of most recent frames. Every time a new frame is processed the queue gets updated. The new frame goes into the queue and the oldest frame gets flushed out.

As already described, the frame queue is used to determine which frame will be copied to the output directory.

For normal operation where there is only one transition frame, the program picks previous\_pic\_name2 for copying. For missed frames the queue entry that gets picked depends on the frame length. If for example, frame length is 4, the code on the previous page will pick previous\_pic\_name\_4 from the queue. With frame counter at 6 (frame\_length\_and\_half = 6) the frame that gets picked is 4 frames behind which will be half way between two transitions.

311	<pre>for(i=0; i &lt; strlen(input line); i++)</pre>
312 🗖	-
313	previous_pic_name15[i] = previous_pic_name14[i];
314	previous pic name14[i] = previous pic name13[i];
315	previous pic name13[i] = previous pic name12[i];
316	previous pic name12[i] = previous pic name11[i];
317	previous pic name11[i] = previous pic name10[i];
318	previous pic name10[i] = previous pic name9[i];
319	previous pic name9[i] = previous pic name8[i];
320	previous pic name8[i] = previous pic name7[i];
321	previous pic name7[i] = previous pic name6[i];
322	previous pic name6[i] = previous pic name5[i];
323	previous pic name5[i] = previous pic name4[i];
324	previous pic name4[i] = previous pic name3[i];
325	previous pic name3[i] = previous pic name2[i];
326	previous pic name2[i] = previous pic name1[i];
327	<pre>previous pic name1[i] = input line[i];</pre>
328 -	}

Unfortunately, I cannot explain all of the details in here due to the book size limit. See the complete listings at: http://dl.dropbox.com/u/5667638/tiffdiff.zip

and study it. You can also drop me a note if you have a hard time understanding the code.